



Beispieldokumentation Sample documentation

**Modbus RTU Client für
Stromwandler STWA4MH von Ziehl**

**Modbus RTU Client for current
transformer STWA4MH by Ziehl**

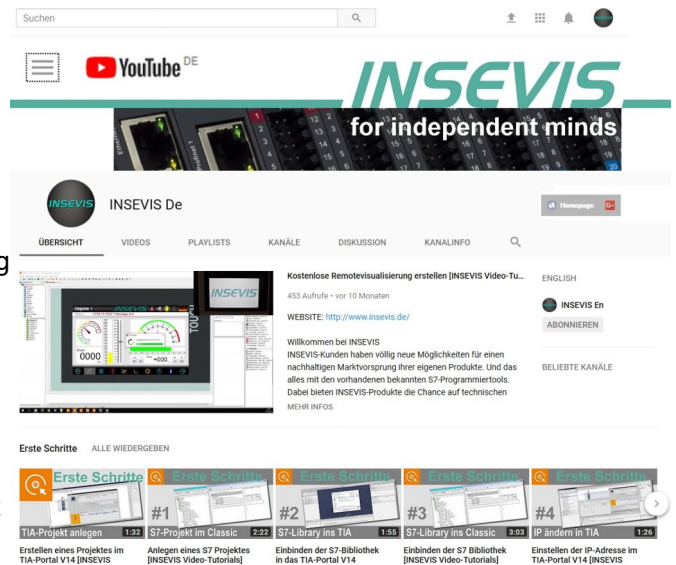
Hinweis zum besseren Verständnis durch Zusatzinformationen

Im deutschen INSEVIS-YouTube-Kanal INSEVIS DE stehen mehrere Playlists mit **Hantierungsvideos** für einzelne Details zur Verfügung.

Ebenfalls stehen **Handbücher** für die einzelnen Produktgruppen im Downloadbereich der Webseite insevis.de zur Verfügung

Bitte nutzen Sie diese Informationsquellen in Ergänzung zur vorliegenden Dokumentation. So können Sie sich noch leichter mit den INSEVIS-Funktionen vertraut machen.

Möchten Sie Erweiterungswünsche oder Fehler zu diesen Beispielen melden oder wollen Sie anderen eigene Beispielprogramme kostenlos zur Verfügung stellen? Gern werden Ihre Programme -auf Wunsch mit Benennung des Autors- allen INSEVIS- Kunden zur Verfügung gestellt.



Hinweis zu den verschiedenen Versionen der Beispielprogramme

Im Lieferumfang der Beispielprogramme können sich auch ältere Ausgabestände bzw. Versionen befinden. Diese wurden nicht aktualisiert und auf die neueste Siemens-Programmiersoftware angepasst, um einen Zugriff mit älteren Programmiersystemen weiterhin zu ermöglichen. Generell werden INSEVIS-Beispielprogramme immer mit dem aktuell neuesten Siemens-Programmierertools erstellt.

BEISPIELBESCHREIBUNG

Inhalt

Das Beispielprogramm realisiert das Auslesen von Messdaten sowie das Schreiben einer Konfiguration zweier STWA4MH Stromwandler über Modbus RTU. Bei den Wandlern sind je 2 Gruppen von Modbus-Registern (aktuelle Messdaten und gepufferte Messdaten) auslesbar, die wahlweise zu- und abschaltbar sind. In der Konfiguration können Modbus-Kommunikationsparameter angepasst und an die Wandler übertragen werden.

Vorgehensweise

- Laden der S7 Software in Client SPS
- Laden der Konfiguration für Modbus-RTU-Client in Client SPS → RUN
- ggf. Laden von S7 Software und Konfiguration für Modbus-RTU-Observer in Observer SPS → RUN
- Beobachten/Steuern über Variablentabellen
 1. „VAT_CFG_Control“ öffnen und Konfigurationsdaten im DB1 einsehen:
 - Slave-Adressen und Transformer-Ratio
 - Baud / Parität / Stoppbits
 2. EINEN Wandler anschließen und innerhalb von 60 Sekunden das Konfigurations-Requestflag für den entsprechenden Wandler setzen:

DB1.DBX	18.0	"STWA4_Instance".Set_Cfg_A	oder
DB1.DBX	18.1	"STWA4_Instance".Set_Cfg_B	
 3. Bei erfolgreicher Konfiguration wird das Requestflag binnen weniger Kommunikationszyklen zurückgesetzt.
 4. Nach erfolgter Konfiguration werden nun alle 500 ms abwechselnd die aktuellen Messdaten der beiden Wandler ausgelesen
 5. Optional können auch vorhergehende Messdaten aus dem Ringpuffer ausgelesen werden, dazu muss jedoch der Aufruf im OB1 angepasst werden.

Hinweis zur STWA4MH-Konfiguration:

Beim Wechsel der Baudrate wird diese vom Knoten erst mit dem nächsten Powercycle übernommen.

S7-Programm

Das Betriebssystem stellt Sende- und Empfangsfunktionen zur Modbus-RTU-Kommunikation bereit. Diese beinhalten die Rahmensynchronisation nach Modbus-Spezifikation einschließlich Checksummenberechnung. Als Interface zur Applikation dient FB2:

```
CALL  "RTUClient"                // FB2 als Multiinstanz
  R      :=FALSE                 // Reset-Eingang, einmalig nach Hochlauf setzen
  UID    :=B#16#1                // Geräteadresse
  Cmd    :=B#16#3                // Modbus Kommando (Funktionscode 1,2,3,4,6,15,16)
  Index  :=16                    // Register bzw. Bit-Adresse (0...65535)
  Length :=11                    // Anzahl zu übertragender Register (1..125) bzw. Bits (1..2000)
  Data   :="STWA4_Data".Head_A  // ANY-Pointer auf Nutzdatenbereich
  Done   :="RTU_done"           // TRUE wenn erfolgreich
  Error  :="RTU_error"          // TRUE bei Fehler
  ErrSrc :="RTU_ErrorSrc"       // Fehlercode s. Tabelle
  ErrStatus:="RTU_ErrCode"
```

Daraus werden in FB2 die Telegramme erstellt, gesendet, empfangen und die Nettodaten kopiert. Für Modbus-Header und Checksumme besteht ein eigener Puffer im Instanzdatenbaustein. Es erfolgt ebenfalls eine Timeout-Überwachung. Erfolgt keine Antwort innerhalb 1 Sekunde, wird der Empfang abgebrochen und das Errorflag gesetzt. Werden CRC-Fehler erkannt oder wird ein RTU-Request vom Server zurückgewiesen, wird ebenfalls das Errorflag gesetzt, der zurückgelesene Funktionscode wird überprüft, weitere Protokollfehler werden nicht beachtet.

STWA4MH

Da keine Schalter für Geräte-IDs vorhanden sind, starten alle Wandler für 60 s mit der zusätzlichen ID 247.

a) In dieser ersten Minute darf keine Kommunikation mit ID 247 erfolgen ODER

b) Es wird extern sichergestellt, dass nur EIN Wandler mit dem System verbunden ist.

Innerhalb dieser 60 s muss FB1 mit passenden Parametern zyklisch aufgerufen werden, bis das Requestflag sich selbst zurückgesetzt hat.

Implementierung

Im OB1 werden per festem Parameter die Modbus-Slaveadressen, Transformer-Ratios, Kommunikationsparameter (Baudrate etc) sowie die Registergruppen zum Lesen festgelegt.

Nach Bedarf anpassen:

SlaveAddr_...	[1..246]		
TransfRatio_...	[1..10]		
BaudID	0: 4800	1: 9600 (default)	2:19200
	3:57600	4:115200	
Parity	0: no	1: even (default)	2: odd
Stoppbits	[1 (default), 2]		

Da der Instanzdatenbaustein die Zustandsmerker enthält und diese null-initialisiert sein sollten, ist dieser auf „non-retain“ gesetzt und eignet sich daher nicht, die Parameter zu speichern. (Alternativ könnten diese in OB100 gelöscht werden)

Im OB1 wird per Timer die Kommunikation verzögert.

a) um nicht unnötig Ressourcen zu belegen und b) um das debuggen der statemachine beobachtbar zu machen. Der Timeout kann angepasst werden.

Fehlercodes

Die Rückgabewerte des FB2 sind in eine Fehlerquelle (ErrSrc) und einen StatusCode (ErrStatus) aufgeteilt und werden bis in OB1-Ebene hochgereicht. Die Fehlerflags teilen sich alle Funktionen. ErrSrc entspricht dem state der statemachine in FB2, in dem der Fehler aufgetreten ist. Davon abhängig sind die jeweiligen Fehlercodes:

ErrSrc	ErrStatus	Bedeutung
0	8001 _{hex}	RS485 nicht in Betriebsart ModbusRTU
		sonst Rückgabewerte des SFB 61 bei Initialisierung
1	8001 _{hex}	UID > 247
	8002 _{hex}	Ungültiges Modbus-CMD (function code)
	8003 _{hex}	Ungültige Längenangabe (Register > 250, Bits/Coils > 2000)
	8004 _{hex}	ungültiger Bereich Nutzdaten (≠ E, A, M, DB)
2	8000 _{hex}	SFB 60 belegt (Sendepuffer overflow)
		Rückgabewert SFB20 oder SFB 60 (Senden)
3		Rückgabewert SFB 61 bei Empfang
	CAFE _{hex}	Timeout
4	9001 _{hex}	Bit 7 im Empfangswert des Funktionscodes gesetzt = Server weist Anfrage zurück (ungültige Parameter)
	9000 _{hex}	sonstiger Fehler im Empfangswert des Funktionscodes
		Rückgabewert des SFC 20 beim Kopieren der Nutzdaten

Modbus RTU Observer:

Falls kein PC mit einer RS485-Schnittstelle verfügbar ist, kann im Problemfall eine weitere SPS als Protokollbeobachter eingesetzt werden. Alle empfangenen Bytes werden mit einem Zeitstempel in einen Ringpuffer geschrieben. (Das Zeitraster ist mit 10 ms zu grob, um einzelne Telegramme zu trennen, aber die Anfragen des Client können zeitlich beliebig verzögert werden. Die Hardwarekonfiguration über ConfigStage muss als plain ASCII erfolgen.

Konfiguration

Client:

Schnittstelle RS485:

- Baudrate auf 9600
- Datenformat 8E1
- Protokoll Modbus RTU
- Serveraktivierung AUS

Eigenschaft: RS485

Anschlusseinstellung

Baudrate:

Datenformat:

Protokoll:

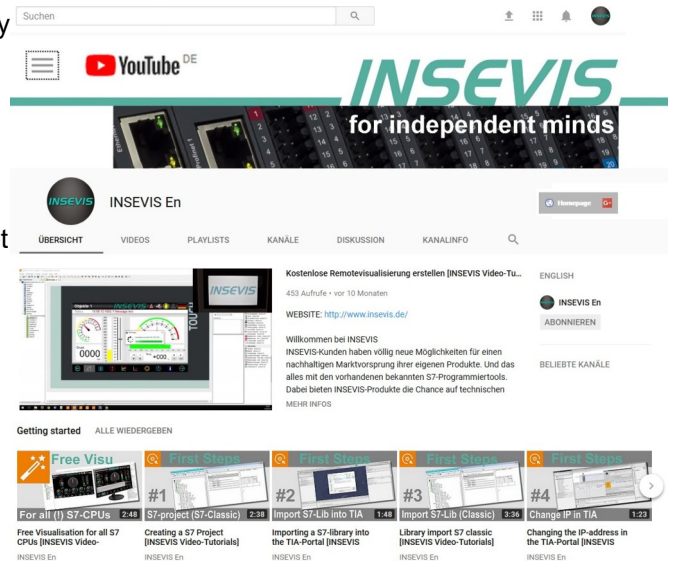
ModBus RTU Serveraktivierung

Hint for better understanding by additional information

In the English YouTube-channel INSEVIS EN we supply different playlists with handling videos for single details. This will help you to get familiar with INSEVIS much faster.

Please download the referring manual from the download area of our English website [insevis.com](http://www.insevis.com) to get familiar with INSEVIS technology in detail.

Do you want to inform us about necessary increments or errors or do you want to provide us with your sample programs to offer it for free to all customers? Gladly we would provide your program -if you wish with the authors name- to all other customers of INSEVIS.



Hint to different versions of the sample programs

There could be older versions in delivery scope of the sample programs too. These were not updated and converted to the newest programming tool versions to allow access by older programming tools too. INSEVIS sample programs will be created in the present newest Siemens-programming tool always.

SAMPLE DESCRIPTION

Abstract

The example program realizes the reading of measurement data as well as the writing of a configuration of two STWA4MH current transformers via Modbus RTU. For each of the transformers, 2 groups of Modbus registers (current measurement data and buffered measurement data) can be read, which can be switched on and off as required. In the configuration, Modbus communication parameters can be adapted and transmitted to the transformers.

Procedure

- transfer S7 software into client PLC
- transfer configuration for Modbus RTU client into client PLC via ConfigStage → RUN
- if needed transfer S7 software and configuration for Modbus RTU observer into observer PLC → RUN
- monitor and control via watch tables
 1. open „VAT_CFG_Control“ and supervise configuration data in DB1
 - slave addresses and ratio values
 - Baud / Parität / Stoppbits
 2. connect ONE transformer and within 60 seconds set the configuration request flag for the corresponding converter in DB1:

DB1.DBX	18.0	"STWA4_Instance".Set_Cfg_A	or
DB1.DBX	18.1	"STWA4_Instance".Set_Cfg_B	
 3. if the configuration is successful, the request flag is reset within a few communication cycles
 4. after successful configuration, the current measurement data of the two transducers are read out alternately every 500 ms
 5. optionally, previous measurement data can also be read out of the ring buffer, however, the call in OB1 must be adapted for this

Note on STWA4MH configuration:

If the baud rate is changed, the node will not adopt it until the next power cycle.

S7 program

The operating system supports Modbus-RTU communication by low level send- and receive functions. These handles frame synchronisation and checksum calculation regarding Modbus specification.

The S7 code FB2 works as application interface:

```

CALL  "RTUClient"                // FB2 as Multiinstance
  R      :=FALSE                 // Reset-input, to set once while startup
  UID    :=B#16#1                // node address
  Cmd    :=B#16#3                // Modbus command (function code 1,2,3,4,6,15,16)
  Index  :=16                    // Register rsp. Bit-address (0...65535)
  Length :=11                    // number of register (1..125) or bits (1..2000) to transfer
  Data   :="STWA4_Data".Head_A  // ANY-Pointer payload data area
  Done   :="RTU_done"           // TRUE if well done
  Error  :="RTU_error"          // TRUE in case of trouble
  ErrSrc :="RTU_ErrorSrc"       // error code s. table
  ErrStatus:="RTU_ErrCode"

```

FB2 builds the telegramm, sends and receives and copies data into specified user area. Buffer for Modbus header and checksum is in FB2's instance datablock. A timeout of 1 second cancels receive and signals by errorflag. In case of detected CRC-errors or denied RTU-requests the errorflag is set too. The received function code is compared to the sended code, further invalid protocoll data are not handled.

STWA4MH

Since there are no switches for device IDs, all converters starts for 60 s with the additional ID 247.

a) No communication with ID 247 is allowed in this first minute OR

b) It is ensured externally that only ONE converter is connected to the system.

Within these 60 s FB1 must be called cyclically with suitable parameters until the request flag has reset itself.

Implementation

In OB1, the Modbus slave addresses, transformer ratios, communication parameters (baud rate, etc.) and the register groups for reading are defined by fixed parameters.

Adapt as required:

SlaveAddr_...	[1..246]		
TransfRatio_...	[1..10]		
BaudID	0: 4800	1: 9600 (default)	2:19200
	3:57600	4:115200	
Parity	0: no	1: even (default)	2: odd
Stoppbits	[1 (default), 2]		

Since the instance data block contains the states of the statemachine and these should be zero initialized, it is set to "non-retain" and is therefore not suitable for saving the parameters.

(Alternatively these could be initialized in OB100)

In OB1 the communication is delayed by a timer.

a) to avoid using unnecessary resources and b) to make the debugging of the statemachine observable.

The timeout can be adjusted.

Error codes

Return values of FB2 are divided into error source (ErrSrc) and a status code (ErrStatus).

Error source accords with the last state of the state machine in FB2, as the error occurred. Related to the error source the status code contains information about the cause of error:

ErrSrc	ErrStatus	description
0	8001 _{hex}	RS485 not configured as ModbusRTU
		else Status return of SFB 61 while initialization
1	8001 _{hex}	UID > 127
	8002 _{hex}	Invalid CMD (function code)
	8003 _{hex}	Invalid LEN (register > 250, bits/coils > 2000)
	8004 _{hex}	Invalid area of payload data (≠ I, O, M, DB)
2	8000 _{hex}	SFB 60 busy (send buffer overflow)
		else Status return SFB20 or SFB 60
3		Status return SFB 61 while receive
	CAFE _{hex}	Timeout
4	9001 _{hex}	Bit 7 of returned function code set = Server denies request (invalide parameter)
	9000 _{hex}	Returned function code invalid
		Status return of SFC 20 copying user data

Modbus RTU Observer:

For troubleshooting and in case no PC with a RS485-UART interface is available, another PLC can be used to observe running communication. All received bytes are stored with a timestamp into a ringbuffer. The time resolution of 10ms is too coarse to detect single frames but the requests by the client are apart enough. Notice that the hardware must be be configured by ConfigStage as plain ASCII.

Configuration

Client:

RS485 interface:

- Baud rate 9600
- Data format 8E1
- Protocol Modbus RTU
- Server Enable OFF

Property: RS485

Port setting

Baud rate:

Data format:

Protocol:

ModBus RTU Server Enable

INSEVIS Vertriebs GmbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail: info@insevis.de

NUTZUNGSBEDINGUNGEN

Die Verwendung der Beispielprogramme erfolgt ausschließlich unter Anerkennung folgender Bedingungen durch den Benutzer: INSEVIS bietet kostenlose Beispielprogramme für die optimale Nutzung der S7-Programmierung und zur Zeitersparnis bei der Programmerstellung. Für direkte, indirekte oder Folgeschäden des Gebrauchs dieser Software schließt INSEVIS jegliche Gewährleistung genauso aus, wie die Haftung für alle Schäden, die aus die aus der Weitergabe der die Beispielinformationen beinhaltenden Software resultieren. Mit Nutzung dieser Dokumentation werden diese Nutzungsbedingungen anerkannt.

TERMS OF USE

The use of this sample programs is allowed only under acceptance of following conditions by the user:
The present software is for guidance only aims at providing customers with sampling information regarding their S7-programs in order to save time. As a result, INSEVIS shall not be held liable for any direct, indirect or consequential damages respect to any claims arising from the content of such software and/or the use made by customers of this sampling information contained herein in connection with their own programs.
Use of this documentation constitutes acceptance of these terms of use.